AD-A085 027 DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 9/2
THE STAGING TEXT EDITOR/DATA BASE LOADER PROGRAM MANUAL.(U)
MAY 80 J M MCKEE
UNCLASSIFIED DTNSRDC-80/072 NL

END
DATE FILMED
7-80
DTIC

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER DTNSRDC-80/072	2. GOVT ACCESSION NO. AD-A085 027	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) THE STAGING TEXT EDITOR/DATA BASE LOADER PROGRAM MANUAL.		5. TYPE OF REPORT & PERIOD COVERED Final 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) James M. McKee		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS David W. Taylor Naval Ship Research and Development Center Bethesda, Maryland 20084		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Work Unit 1844-122
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE May 1980 13. NUMBER OF PAGES 62
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer Programs
Data Base Management
Finite Element Method
Interactive Graphics
Structural Analysis
Text Editor

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes programs and procedures which can be used to create an interactive graphics data base from the results of finite element structural analysis programs and similar computer programs. Finite element models and the results of computations, stored in this form, can be displayed and manipulated using interactive graphics terminals. One program is an interactive text editor for reformatting the print file produced by the

(Continued on reverse side)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(Block 20 continued)

analysis programs. A second program carries out the data base creation and modification operations. All program commands and controls are described and illustrated by examples.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This report describes programs and procedures which can be used to create an interactive graphics data base from the results of finite element structural analysis programs and similar computer programs. Finite element models and the results of computations, stored in this form, can be displayed and manipulated using interactive graphics terminals. One program is an interactive text editor for reformatting the print file produced by the analysis programs. A second program carries out the data base creation and modification operations. All program commands and controls are described and illustrated by examples.

ADMINISTRATIVE INFORMATION

This project was funded under Military Interdepartmental Purchase Request number FY 14567900016 issued by the Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio. The work was performed under Work Unit 1844-122.

INTRODUCTION

STAGING is an interactive support system for finite element analysis which runs on large CDC computers. STAGING provides tools for creating and modifying finite element models and for graphic display of the model and the results of analysis runs like the views of a 3-D solid finite element model shown in Figure 1.[1*] Before the development of the text editor and data base loader, STAGING users were required to construct interface programs to link their particular analysis programs with STAGING. The STAGING text editor and loader provides a means for using STAGING to display the finite element model and analysis results without the use of specially tailored interface programs.

The STAGING text editor and loader can be used with any computer data available in character format. The data input file and the printer output file associated with an analysis run are usually the most

*A complete listing of references is given on page 55.

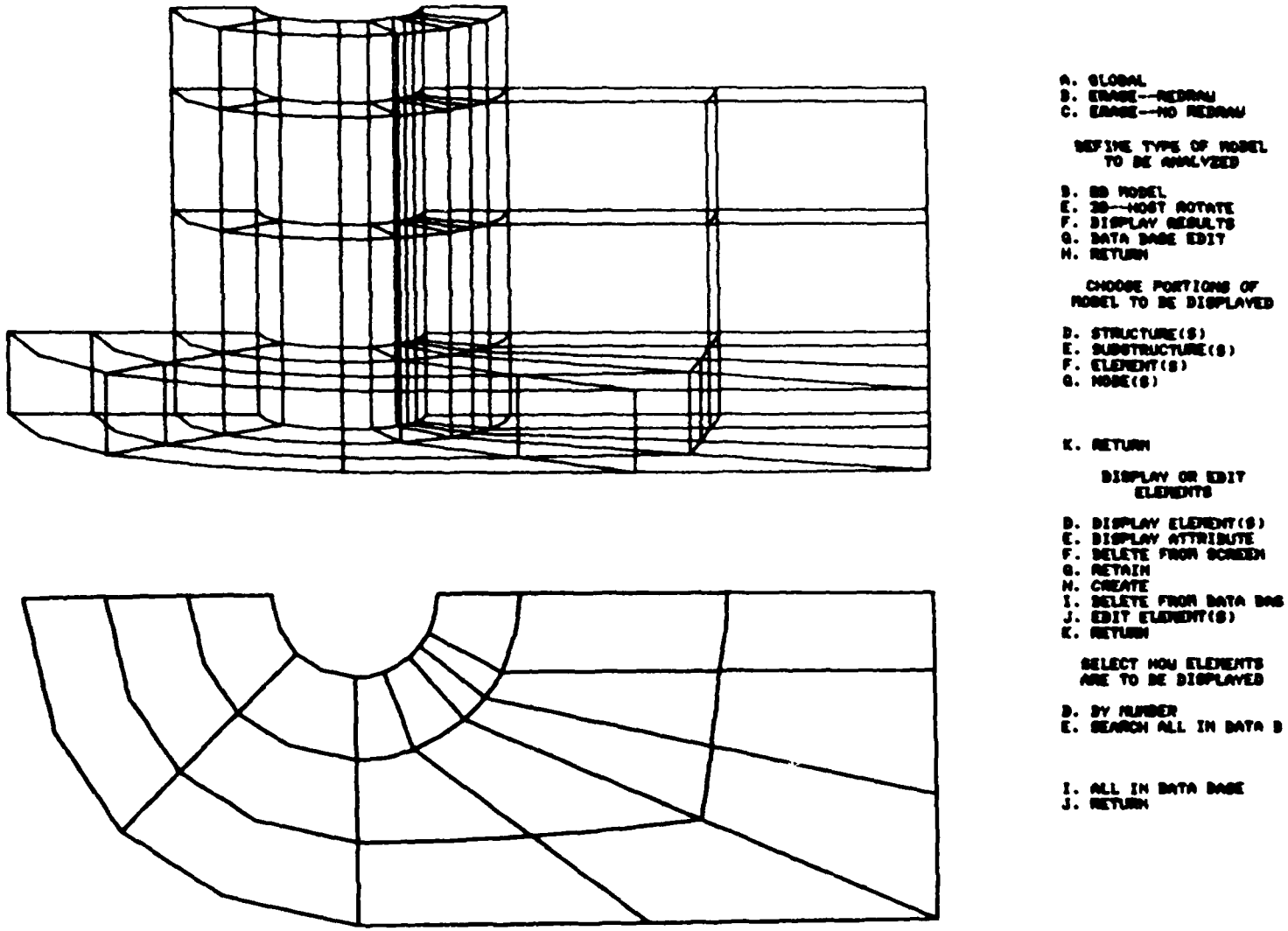


Figure 1 - STAGING Views of a 3-D Solid Finite Element Model

convenient source of data for the text editor and loader. In order to use the printer output file with the editor and loader, the user must make provisions for saving that file on permanent disk storage. The text editor and the data base loader can be used in either interactive or batch modes. Editor and loader commands can also be saved and used repetitively for a series of similar analysis problems.

Like STAGING, the text editor and the data base loader will run on most large CDC computers. Both the text editor and the loader are written in ANSI standard FORTRAN IV. No special assembler language subroutines have been written and the only such routines used are part of the STAGING data base support library. The language constraints imposed on the programmer should simplify the conversion of the programs for use on other computers. However, such constraints have impaired the efficiency of the text editor to some extent and have limited the file manipulation capability within the text editor. It should be noted that, although the text editor and loader provide a comprehensive data base loading capability, they do not provide a means for extracting data from a STAGING data base. The use of the text editor with existing data extraction programs offers a much broader capability in this area.

USING THE PROGRAMS

The STAGING data base loader processes a file consisting of loader commands and numeric data to produce a new or updated STAGING data base. The loader commands are used to specify the relationship between the numeric data and the various categories and identifiers used in the data base organization. Any file containing commands and data which conforms to the data base loader rules may be used as loader input. The STAGING text editor is simply one tool for transforming a data file in a user's format to an acceptable data base loader file.

THE TEXT EDITOR

The STAGING text editor is an implementation of the text editor described by Kernighan and Plauger in their book Software Tools.[2] This editor has many features which simplify the task of producing a loader

file. This section of the manual has been limited to a discussion of some basic editing operations which are illustrated with simple examples. The user should refer to Appendix A for a complete description of the editor commands and operations.

To begin editing a file the user enters

```
EDIT, <file>
```

where <file> is the name of the file containing text to be edited.

The editor is basically line oriented and the range of each editor command is controlled by a set of line numbers. The command

```
1 P
```

will cause the first line to be printed and the command

```
1,3 D
```

will cause lines 1, 2, and 3 to be deleted.

A period (.) is used to indicate the current line and the dollar sign ($) refers to the last line of the file. Simple arithmetic may be performed to modify the line numbers. Thus

```
.+1,$-1 D
```

will delete all lines between the current line and the last line.

The editor will determine the line numbers for lines containing a specified text pattern whenever it encounters a text string bracketed by slashes in place of a line number. Forward slashes (/) specify the first line after the current line which contains the specified pattern. Similarly, backward slashes (\) specify the first previous line which contains the pattern. Such pattern searches are performed in a circular mode which proceeds to the first line after the last line has been passed (the opposite holds for backward searches). Thus

```
\BEGIN BULK\,/ENDDATA/ P
```

will print the first previous line containing "BEGIN BULK" and all subsequent lines through the next line containing "ENDDATA".

When more than one line number is required, the line numbers are usually separated by commas. If line numbers are separated by semicolons, the preceding line number will be established as the current line number. If more line numbers are specified than the maximum permitted for the command, the rightmost line numbers will be used. The command

```
/ABC/,/ABC/ P
```

will print the next line containing "ABC" as would

```
/ABC/ P or /ABC/;. P
```

The command

```
/ABC/;/ABC/;.,/ABC/ P
```

will print the second line containing "ABC" and all subsequent lines through the third line containing "ABC".

The editor "remembers" the last search pattern specified so that the previous command could have also been given as

```
/ABC/;//;.,// P .
```

Basic editor commands are one character long and blank characters can be freely interspersed between command parameters, although none are required. If no command is included with the line numbers, the line indicated by the rightmost number will be printed.

Probably the most heavily used command is "C"; indicating a change or substitution of a text string within a line. The command

```
C/ABC/DEF/
```

will substitute "DEF" for the first occurrence of "ABC" in the line. The substitution can be repeated throughout the line by appending "G" (for global) at the end of the command. Thus

```
C/ABC/DEF/ G
```

will substitute "DEF" for all occurrence of "ABC" within a line.

The results of an editor operation will be printed if a "P" is appended to the command. Thus

```
C/ABC/DEF/ G P
```

will display the line after the substitution of the previous example has been made.

If a string contains a slash, the user may choose other string delimiters or precede the slash by the escape character, (@). The commands

```
C/@//-/ and
C;/;-;
```

both change a slash to a dash.

Many options are available for forming search and substitution patterns. A "!" is used to indicate the beginning of a line and "$" indicates the end of a line. "?" represents any character and "*" indicates zero or more occurrences of the previous character. Lines can be divided by inserting "@N"s to indicate where new lines are to begin. An "&" is used in a substitution string to indicate the matched pattern. Thus

```
C/AGAIN/& AND &/
```

will change "AGAIN" to "AGAIN AND AGAIN".

The pattern matching capability of the text editor is quite powerful. The pattern

```
/ABC???/
```

will match all strings which begin with "ABC" followed by any three additional characters. The pattern

```
/ABC??*/
```

will match all strings beginning with "ABC" which are followed by one or more additional characters. The pattern

```
/ABC{0-9}{0-9}*/
```

will match "ABC" followed by one or more digits. The pattern

```
/ABC{^A-Z}/
```

will match "ABC" followed by any character which is not alphabetic.

Text editor commands can be applied to several lines of text by specifying a global range for the commands. The command

```
1,$ G /ERROR/ P
```

will print all lines which contain the string "ERROR". The command

```
$-99,$ X /! / = P
```

will print the line and the line number (requested by "=") for all lines within the last 100 lines of the file, excluding (from the "X") those lines that begin with a blank character.

If ";" is chosen as a tab character, the global change command

```
G /;/ C // / G
```

will cause the text to be indented five spaces wherever the tab character occurs. Similarly, the two commands

```
G/??????????????????????????????????????/ C//&@NTRASH/
G/!TRASH/ D
```

will truncate all lines to 40 characters.

The command

```
I
```

permits the user to insert lines of text after the current line. Text insertion will terminate when a line is entered which contains only a period (.). The command

```
R FILEA
```

causes the editor to read FILEA and insert those lines read after the current line. Lines of text can be saved on a file using the write command, "W". Thus

```
.,$ W FILEB
```

will write the current line and the remainder of the edit file to a file named "FILEB". All files will be rewound before and after read, write, and edit ("EDIT, <file>" and "E") commands with the exception of the file SINPUT which is not rewound when a write command is processed and following a read command. The special nature of the SINPUT file permits the user to accumulate text on that file.

The user is cautioned that the file names used by the text editor must either be listed on the "EDIT" system control command or be one of the default file names listed in Appendix A. Further, the user must not attempt to overwrite a permanent disk file. This will cause the immediate termination of the text editor and the loss of the current edit file. The system command sequence given in Appendix B may be used to avoid most of the problems associated with the creation and editing of permanent disk files.

THE DATA BASE LOADER

The STAGING data base loader reads a file of commands and data and either creates a new STAGING data base or appends information to an existing STAGING data base. Data base creation and modification are accomplished using STAGING's data base access subroutines. This section of the manual has been limited to a general discussion of the operation of the loader. The user should refer to Appendix C for a complete

description of the loader commands and operations and to the EDITOR/LOADER EXAMPLE section for some typical procedures used to create and modify STAGING data bases.

Data to be loaded by the loader must be grouped in sets which have common characteristics. The command

SET, NODES

defines the beginning of a set of node-type data to be stored in the data base. The command

SET, ELEMENTS

similarly refers to a set of element-type data to be stored in the data base.

To further specify the characteristics of the data, the commands: "VARIABLE", "CONSTANT", and "INDEX" may be used. Various combinations of these commands permit the user to specify the attributes and format of data records to be loaded. A data record consists of a fixed number of data fields which may be spread over several input lines. A data field can be an integer number, a real number, or null. The data fields on one input line must be separated by blanks or commas.

The location and attributes of data that will be read explicitly from the input file are defined using the VARIABLE command. Thus

VARIABLE, NAME, SKIP, X-CORD, Y-CORD

specifies a four-field data record for node-type data. The first field is to contain the node ID or NAME, the second field is to be ignored, and the third and fourth fields are to contain x- and y-coordinate data, respectively. Additional variable commands, within the same set, can be used to specify fields in addition to the four current fields.

The values of data attributes that will remain constant for all data records in the current set are specified using the CONSTANT command. Thus

CONSTANT, CORD-SYSTM, 1, Z-CORD, 1.5

specifies that each data record in this set will have a coordinate system attribute of "1" and a z-coordinate attribute of "1.5" stored with the data that is read explicitly. The CONSTANT command is also used to define initial values for attributes that are to be incremented. If the

same attribute is included as part of more than one CONSTANT command, the last value given will be used for data loading.

The INDEX command is used to define an increment for an attribute value or an attribute subscript that is to be applied after each data record is processed. The sequence

```
VARIABLE, SKIP, NODE, NODE, NODE
CONSTANT, NAME, 100, TYPE, $GEN TRIA$
INDEX, NAME, 10
```

could be used to specify element ID's for triangular elements. The ID's 100, 110, 120, ... will be generated by this sequence. The sequence

```
VARIABLE, $X DISP$, $Y DISP$
CONSTANT, NAME, 72, $X DISP$, 0, $Y DISP$, 0, $Z DISP$, 0
INDEX, $X DISP$, 1, $Y DISP$, 1, $Z DISP$, 1
CONSTANT, $Z DISP$, .001
```

specifies that the data in this set applies to node 72, that data for several load cases or time steps are to be loaded starting with case zero, and that the z-displacement for each case is not to be read, but instead, set to a value of .001.

The attribute names and element types currently recognized by STAGING are listed in Appendix D. Note that attribute names and element types which contain blanks must be bracketed by a pair of dollar signs. The loader will terminate with an error condition if unknown attribute names or element types are encountered.

The beginning of a set of data records is indicated by the "BEGIN DATA" command. The end of a set is marked by an "ENDDATA" command. Each new data record begins on a new input line. Thus the sequence

```
BEGIN DATA
1,2,3,4
5,6,7,8
9,10,11,12
13,14,15,16
ENDDATA
```

could be used to define data for: four data sets which are up to four fields long, two data sets which are between five and eight fields long, or one data set which is between 13 and 16 fields long.

The commands

```
SET, NODES and
SET, NODES, APPEND
```

are equivalent. They indicate that node-type data are to be appended to an existing data base. In this mode, the user needs only to specify the attributes necessary for defining the data to be loaded in this set. The command

```
SET, NODES, INITIAL
```

indicates that a new data base is to be created. In this mode all data attributes must be defined prior to the loading of any data. The following sequence could be used to define elements and nodes in a new data base:

```
SET, NODES, INITIAL
VARIABLE, NAME, X-CORD, Y-CORD, Z-CORD
CONSTANT, CORD-SYSTM, 1
SET, ELEMENTS, INITIAL
VARIABLE, NAME, NODE, NODE
CONSTANT, TYPE, ROD
BEGIN DATA
<data records>
ENDDATA
SET, NODES, INITIAL
VARIABLE, NAME, X-CORD, Y-CORD, Z-CORD
CONSTANT, CORD-SYSTM, 1
BEGIN DATA
<data records>
ENDDATA
```

Append-mode data sets may follow the initial sets if other data is to be inserted in the same loader run. Append-mode loading permits new attributes to be appended to existing data and new elements and node data to be inserted into the data base. Additional node points can also be appended to element definitions in this mode. It should be noted that initial-mode loading is much more efficient than append-mode loading and should be used whenever possible.

The command and data file will be read until no more commands are found or until an ENDDATA command is encountered outside of a BEGIN DATA/ENDDATA sequence.

Once a command and data file has been assembled and the program file has been attached as file "SLOAD", the loader can be invoked by the system command

```
SLOAD,<infile>
```

where <infile> is the name of the loader command and data file (the

default name for this file is "SINPUT"). About 50000 (octal) words of memory are required for the loader. The data base file is assumed to reside on a file named "TAPE0". For append-mode runs, the file TAPE0 should initially contain a copy of an existing data base. For initial-mode runs, the system control sequence

```
FETCH,SLOAD,CAMK.
ATTACH,SINPUT,MYINPUT,ID=...
SLOAD,SINPUT
CATALOG,TAPE0,MYDATABASE,ID=...
```

will run the loader and save the data base on a permanent disk file. Similarly, for an append-mode run, the system control sequence

```
FETCH,SLOAD,CAMK.
ATTACH,SINPUT,MYINPUT,ID=...
ATTACH,OLDDB,MYDATABASE,ID=...
REQUEST,TAPE0,*PF.
COPYBF,OLDDB,TAPE0.
SLOAD,SINPUT.
CATALOG,TAPE0,MYNEWDATABASE,ID=...
```

will run the loader and save an updated copy of the previous data base on a permanent disk file.

EDITOR/LOADER EXAMPLE

To introduce the prospective user to the techniques that are necessary for running the STAGING text editor and the STAGING data base loader, a complete EDITOR/LOADER example will be discussed in this section. This example begins with an analysis program run and includes the subsequent procedure steps that are necessary to display the model and the analysis results with the STAGING program. Because of the flexibility of the editor and loader, any example will be just one of many possible procedure sequences that will lead to the same results. The choice of procedures is determined primarily by the format of the data produced by the analysis program. The editor is used to transform the format given by the analysis program into one that can be recognized by the loader. This includes the insertion of header cards to properly identify the various types of data. In keeping with the order established elsewhere in this report, the editor example will be discussed before the loader example. The reader is encouraged to page forward to the loader

example whenever he needs clarification of the author's reasons for choosing a particular editing operation.

A FINITE ELEMENT ANALYSIS PROBLEM

The complete printer output file produced by a static analysis run using the NASTRAN finite element structural analysis program[3] is reproduced on the pages that follow. (Note that the printer control characters which govern pagination and line spacing have also been included in this listing of the file.) For this type of run NASTRAN requires that node-type data be given on "RINGAX" cards, that conical finite elements be defined on "CCONEAX" cards, and that loading data be given on "PRESAX" cards. Displacement results have been calculated for all nodes, but the results are nonzero for only the zeroeth harmonic. Stress results have been calculated only for elements 15 and 35.

0

```
NASTRAN SYSTEM(76)=1
1
0
```

0 N A S T R A N E X E C U T I V E C O N T R O L D E C K E C H O
0

```
ID NASTRAN,DEMO
APP DISP
```

```
   SOL 1,0
   TIME 10
   CEND
1  NASTRAN COURSE  - - -  DEMO. PROB. 1B                                  AUGUST  30, 1979   NASTRAN 11/30/78    PAGE      2
   RING-STIFFENED CYLINDER WITH UNIFORM PRESSURE LOAD
0  CONICAL SHELL ELEMENTS  (COMPARE TO PROBLEM 1A, SUBCASE 3)
0                                C A S E    C O N T R O L    D E C K    E C H O

                CARD
                COUNT
                  1      TITLE=NASTRAN COURSE  - - -  DEMO. PROB. 1B
                  2      SUBTITLE=RING-STIFFENED CYLINDER WITH UNIFORM PRESSURE LOAD
                  3      LABEL=CONICAL SHELL ELEMENTS  (COMPARE TO PROBLEM 1A, SUBCASE 3)
                  4      AXISYM=COSINE
                  5      SET 7 = 15,35
                  6      LOAD=15
                  7      HARMONICS=ALL
                  8      DISP=ALL
                  9      STRESS=7
                 10      SPCFORCE=ALL
                 11      OLOAD=ALL
                 12      BEGIN BULK
0
1  NASTRAN COURSE  - - -  DEMO. PROB. 1B                                  AUGUST  30, 1979   NASTRAN 11/30/78    PAGE      3
   RING-STIFFENED CYLINDER WITH UNIFORM PRESSURE LOAD
0  CONICAL SHELL ELEMENTS  (COMPARE TO PROBLEM 1A, SUBCASE 3)
0                                S O R T E D    B U L K    D A T A    E C H O

                CARD
                COUNT        .  1  ..  2  ..  3  ..  4  ..  5  ..  6  ..  7  ..  8  ..  9  .. 10  .
                  1-         AXIC    2
                  2-         CCONEAX 2       4       2       3
                  3-         CCONEAX 3       4       3       5
                  4-         CCONEAX 5       3       5       10
                  5-         CCONEAX 10      3       10      15
                  6-         CCONEAX 15      3       15      20
                  7-         CCONEAX 20      3       20      25
                  8-         CCONEAX 25      3       25      30
                  9-         CCONEAX 30      3       30      35
                 10-         CCONEAX 35      3       35      40
                 11-         CCONEAX 40      3       40      45
                 12-         CCONEAX 42      4       42      43
                 13-         CCONEAX 43      4       43      45
                 14-         CCONEAX 45      3       45      50
                 15-         CCONEAX 50      3       50      55
                 16-         CCONEAX 55      3       55      60
                 17-         CCONEAX 60      3       60      65
                 18-         MAT1    4       3.+7            .3      7.324-4
                 19-         PCONEAX 3       4       1.      4       .08333  4       1.              +PCON1
                 20-         +PCON1  -0.5    0.5     0.      30.     60.     90.     135.    180.
                 21-         PCONEAX 4       4       .5      4       .010417 4       .5
                 22-         POINTAX 36      35      45.
                 23-         PRESAX  15      -1.0    5       10      .0      360.
                 24-         PRESAX  15      -1.0    10      15      .0      360.
                 25-         PRESAX  15      -1.0    15      20      .0      360.
                 26-         PRESAX  15      -1.0    20      25      .0      360.
                 27-         PRESAX  15      -1.0    25      30      .0      360.
                 28-         PRESAX  15      -1.0    30      35      .0      360.
                 29-         PRESAX  15      -1.0    35      40      .0      360.
                 30-         PRESAX  15      -1.0    40      45      .0      360.
```

```
                31-        PRESAX  15      -1.0    45      50      .0      360.
                32-        PRESAX  15      -1.0    50      55      .0      360.
                33-        PRESAX  15      -1.0    55      60      .0      360.
                34-        PRESAX  15      -1.0    60      65      .0      360.
                35-        RINGAX  2               57.5    .0                      46
                36-        RINGAX  3               58.75   .0                      46
                37-        RINGAX  5               60.     .0                      4
                38-        RINGAX  10              60.     4.                      4
                39-        RINGAX  15              60.     8.                      4
                40-        RINGAX  20              60.     12.                     4
                41-        RINGAX  25              60.     16.                     4
                42-        RINGAX  30              60.     20.                     4
                43-        RINGAX  35              60.     24.                     4
                44-        RINGAX  40              60.     28.                     4
                45-        RINGAX  42              57.5    32.                     46
                46-        RINGAX  43              58.75   32.                     46
                47-        RINGAX  45              60.     32.                     4
                48-        RINGAX  50              60.     36.                     4
                49-        RINGAX  55              60.     40.                     4
                50-        RINGAX  60              60.     44.                     4
1  NASTRAN COURSE  - - -  DEMO. PROB. 1B                          AUGUST  30, 1979   NASTRAN 11/30/78    PAGE      4
   RING-STIFFENED CYLINDER WITH UNIFORM PRESSURE LOAD
0  CONICAL SHELL ELEMENTS  (COMPARE TO PROBLEM 1A, SUBCASE 3)
0                                   S O R T E D   B U L K   D A T A   E C H O
              CARD
              COUNT        .  1  ..  2  ..  3  ..  4  ..  5  ..  6  ..  7  ..  8  ..  9  .. 10  .
                51-        RINGAX  65              60.     48.                     3452
                           ENDDATA
0
0       **NO ERRORS FOUND - EXECUTE NASTRAN PROGRAM**


1  NASTRAN COURSE  - - -  DEMO. PROB. 1B                          AUGUST  30, 1979   NASTRAN 11/30/78    PAGE      5
   RING-STIFFENED CYLINDER WITH UNIFORM PRESSURE LOAD
0  CONICAL SHELL ELEMENTS  (COMPARE TO PROBLEM 1A, SUBCASE 3)
0
0
0


0*** USER WARNING MESSAGE 2015, EITHER NO ELEMENTS CONNECT INTERNAL GRID POINT         1
 OR IT IS CONNECTED TO A RIGID ELEMENT OR A GENERAL ELEMENT.


0*** SYSTEM INFORMATION MESSAGE 3113, EMGPRO PROCESSING SINGLE PRECISION ELEMENTS OF TYPE       35 STARTING WITH ID     2001
0*** SYSTEM INFORMATION MESSAGE 3107.  EMGOLD IS PROCESSING ELEMENTS OF TYPE = 35, BEGINNING WITH ELEMENT ID =      2001


                                    MPYAD--NULL MATRIX PRODUCT
                                    MPYAD--NULL MATRIX PRODUCT
                                    MPYAD--NULL MATRIX PRODUCT
0***USER INFORMATION MESSAGE 3023--PARAMETERS FOR SYMMETRIC DECOMPOSITION OF DATA BLOCK KLL      ( N =  205 )
            TIME ESTIMATE=      1       C AVG =     6       PC AVG =     0   SPILL GROUPS =     0          S AVG =     1
         ADDITIONAL CORE= -28431        C MAX =    11        PCMAX =     0      PC GROUPS =     0  PREFACE LOOPS =     1


                                    MPYAD--NULL MATRIX PRODUCT
                                    MPYAD--NULL MATRIX PRODUCT
                                    METHOD 2 NT,NBR PASSES =   1,EST. TIME =      .5
```

0*** USER INFORMATION MESSAGE 3035
OFOR LOAD   1 EPSILON SUB E = -1.0067135E-12

MPYAD--NULL MATRIX PRODUCT
METHOD 2 T ,NBR PASSES = 1,EST. TIME = .1
METHOD 2 NT,NBR PASSES = 1,EST. TIME = .2

0*** USER WARNING MESSAGE 2076, SDR2 OUTPUT DATA BLOCK NO. 1 IS PURGED
0*** USER WARNING MESSAGE 2077, SDR2 OUTPUT DATA BLOCK NO. 2 IS PURGED
0*** USER WARNING MESSAGE 2078, SDR2 OUTPUT DATA BLOCK NO. 3 IS PURGED

RING-STIFFENED CYLINDER WITH UNIFORM PRESSURE LOAD
0 CONICAL SHELL ELEMENTS (COMPARE TO PROBLEM 1A, SUBCASE 3)

D I S P L A C E M E N T V E C T O R

SECTOR-ID / POINT-ID / RING-ID	HARMONIC	T1	T2	T3	R1	R2	R3
36		-1.167751E-04	0.0	-1.372601E-05	0.0	1.112742E-06	0.0
2	0	-9.118936E-05	0.0	-3.630717E-05	0.0	-3.537418E-06	0.0
3	0	-9.062216E-05	0.0	-3.189937E-05	0.0	-3.515412E-06	0.0
5	0	-9.010751E-05	0.0	-2.751779E-05	0.0	-3.495443E-06	0.0
10	0	-1.052826E-04	0.0	-2.556372E-05	0.0	-3.469719E-06	0.0
15	0	-1.161534E-04	0.0	-2.333901E-05	0.0	-1.917110E-06	0.0
20	0	-1.210210E-04	0.0	-2.095868E-05	0.0	-6.292796E-07	0.0
25	0	-1.218593E-04	0.0	-1.852483E-05	0.0	1.275398E-07	0.0
30	0	-1.202955E-04	0.0	-1.609989E-05	0.0	6.374756E-07	0.0
35	0	-1.167751E-04	0.0	-1.372601E-05	0.0	1.112742E-06	0.0
40	0	-1.117314E-04	0.0	-1.143988E-05	0.0	1.272038E-06	0.0
42	0	-1.095846E-04	0.0	-9.245275E-06	0.0	1.175702E-09	0.0
43	0	-1.089030E-04	0.0	-9.246740E-06	0.0	1.168388E-09	0.0
45	0	-1.082845E-04	0.0	-9.248197E-06	0.0	1.161751E-09	0.0
50	0	-1.117000E-04	0.0	-7.056703E-06	0.0	-1.251518E-06	0.0
55	0	-1.165694E-04	0.0	-4.772603E-06	0.0	-1.040575E-06	0.0
60	0	-1.196707E-04	0.0	-2.406617E-06	0.0	-5.029012E-07	0.0
65	0	-1.206553E-04	0.0	0.0	0.0	0.0	0.0
2	1	0.0	0.0	0.0	0.0	0.0	0.0
3	1	0.0	0.0	0.0	0.0	0.0	0.0
5	1	0.0	0.0	0.0	0.0	0.0	0.0
10	1	0.0	0.0	0.0	0.0	0.0	0.0
15	1	0.0	0.0	0.0	0.0	0.0	0.0
20	1	0.0	0.0	0.0	0.0	0.0	0.0
25	1	0.0	0.0	0.0	0.0	0.0	0.0
30	1	0.0	0.0	0.0	0.0	0.0	0.0
35	1	0.0	0.0	0.0	0.0	0.0	0.0
40	1	0.0	0.0	0.0	0.0	0.0	0.0
42	1	0.0	0.0	0.0	0.0	0.0	0.0
43	1	0.0	0.0	0.0	0.0	0.0	0.0
45	1	0.0	0.0	0.0	0.0	0.0	0.0
50	1	0.0	0.0	0.0	0.0	0.0	0.0

55	1	0.0	0.0	0.0	0.0	0.0	0.0
60	1	0.0	0.0	0.0	0.0	0.0	0.0
65	1	0.0	0.0	0.0	0.0	0.0	0.0
2	2	0.0	0.0	0.0	0.0	0.0	0.0
3	2	0.0	0.0	0.0	0.0	0.0	0.0
5	2	0.0	0.0	0.0	0.0	0.0	0.0
10	2	0.0	0.0	0.0	0.0	0.0	0.0
15	2	0.0	0.0	0.0	0.0	0.0	0.0
20	2	0.0	0.0	0.0	0.0	0.0	0.0
25	2	0.0	0.0	0.0	0.0	0.0	0.0
30	2	0.0	0.0	0.0	0.0	0.0	0.0
35	2	0.0	0.0	0.0	0.0	0.0	0.0
40	2	0.0	0.0	0.0	0.0	0.0	0.0
42	2	0.0	0.0	0.0	0.0	0.0	0.0
43	2	0.0	0.0	0.0	0.0	0.0	0.0
45	2	0.0	0.0	0.0	0.0	0.0	0.0
50	2	0.0	0.0	0.0	0.0	0.0	0.0
55	2	0.0	0.0	0.0	0.0	0.0	0.0

D I S P L A C E M E N T V E C T O R

SECTOR-ID / POINT-ID / RING-ID	HARMONIC	T1	T2	T3	R1	R2	R3
60	2	0.0	0.0	0.0	0.0	0.0	0.0
65	2	0.0	0.0	0.0	0.0	0.0	0.0

L O A D V E C T O R

SECTOR-ID / POINT-ID / RING-ID	HARMONIC	T1	T2	T3	R1	R2	R3
5	0	-7.539822E+02	0.0	0.0	0.0	0.0	0.0
10	0	-1.507964E+03	0.0	0.0	0.0	0.0	0.0
15	0	-1.507964E+03	0.0	0.0	0.0	0.0	0.0
20	0	-1.507964E+03	0.0	0.0	0.0	0.0	0.0
25	0	-1.507964E+03	0.0	0.0	0.0	0.0	0.0
30	0	-1.507964E+03	0.0	0.0	0.0	0.0	0.0
35	0	-1.507964E+03	0.0	0.0	0.0	0.0	0.0
40	0	-1.507964E+03	0.0	0.0	0.0	0.0	0.0
45	0	-1.507964E+03	0.0	0.0	0.0	0.0	0.0
50	0	-1.507964E+03	0.0	0.0	0.0	0.0	0.0
55	0	-1.507964E+03	0.0	0.0	0.0	0.0	0.0
60	0	-1.507964E+03	0.0	0.0	0.0	0.0	0.0
65	0	-7.539822E+02	0.0	0.0	0.0	0.0	0.0

F O R C E S O F S I N G L E - P O I N T C O N S T R A I N T

SECTOR-ID / POINT-ID / RING-ID	HARMONIC	T1	T2	T3	R1	R2	R3
65	0	0.0	0.0	-7.363269E-09	0.0	6.279852E+02	0.0

S T R E S S E S I N A X I S - S Y M M E T R I C C O N I C A L S H E L L E L E M E N T S (CCONEAX)

ELEMENT ID.	HARMONIC	POINT ANGLE	FIBRE DISTANCE	STRESSES IN ELEMENT COORD SYSTEM NORMAL-V	NORMAL-U	SHEAR-UV	PRINCIPAL STRESSES (ZERO SHEAR) ANGLE	MAJOR	MINOR	MAXIMUM SHEAR
0 15	0		-5.000000E-01	-1.515142E+01	-6.416098E+01	0.0				
			5.000000E-01	1.508066E+01	-5.509136E+01	0.0				
0 15	1		-5.000000E-01	0.0	0.0	0.0				
			5.000000E-01	0.0	0.0	0.0				
0 15	2		-5.000000E-01	0.0	0.0	0.0				
			5.000000E-01	0.0	0.0	0.0				
0 15		0.0000	-5.000000E-01	-1.515142E+01	-6.416098E+01	0.0	0.0000	-1.515142E+01	-6.416098E+01	2.450478E+01
			5.000000E-01	1.508066E+01	-5.509136E+01	0.0	0.0000	1.508066E+01	-5.509136E+01	3.508601E+01
0 15		30.0000	-5.000000E-01	-1.515142E+01	-6.416098E+01	0.0	0.0000	-1.515142E+01	-6.416098E+01	2.450478E+01
			5.000000E-01	1.508066E+01	-5.509136E+01	0.0	0.0000	1.508066E+01	-5.509136E+01	3.508601E+01
0 15		60.0000	-5.000000E-01	-1.515142E+01	-6.416098E+01	0.0	0.0000	-1.515142E+01	-6.416098E+01	2.450478E+01
			5.000000E-01	1.508066E+01	-5.509136E+01	0.0	0.0000	1.508066E+01	-5.509136E+01	3.508601E+01
0 15		90.0000	-5.000000E-01	-1.515142E+01	-6.416098E+01	0.0	0.0000	-1.515142E+01	-6.416098E+01	2.450478E+01
			5.000000E-01	1.508066E+01	-5.509136E+01	0.0	0.0000	1.508066E+01	-5.509136E+01	3.508601E+01
0 15		135.0000	-5.000000E-01	-1.515142E+01	-6.416098E+01	0.0	0.0000	-1.515142E+01	-6.416098E+01	2.450478E+01
			5.000000E-01	1.508066E+01	-5.509136E+01	0.0	0.0000	1.508066E+01	-5.509136E+01	3.508601E+01
0 15		180.0000	-5.000000E-01	-1.515142E+01	-6.416098E+01	0.0	0.0000	-1.515142E+01	-6.416098E+01	2.450478E+01
			5.000000E-01	1.508066E+01	-5.509136E+01	0.0	0.0000	1.508066E+01	-5.509136E+01	3.508601E+01
0 35	0		-5.000000E-01	-1.008167E+01	-6.019096E+01	0.0				
			5.000000E-01	1.007292E+01	-5.414459E+01	0.0				
0 35	1		-5.000000E-01	0.0	0.0	0.0				
			5.000000E-01	0.0	0.0	0.0				
0 35	2		-5.000000E-01	0.0	0.0	0.0				
			5.000000E-01	0.0	0.0	0.0				
0 35		0.0000	-5.000000E-01	-1.008167E+01	-6.019096E+01	0.0	0.0000	-1.008167E+01	-6.019096E+01	2.505465E+01
			5.000000E-01	1.007292E+01	-5.414459E+01	0.0	0.0000	1.007292E+01	-5.414459E+01	3.210875E+01
0 35		30.0000	-5.000000E-01	-1.008167E+01	-6.019096E+01	0.0	0.0000	-1.008167E+01	-6.019096E+01	2.505465E+01
			5.000000E-01	1.007292E+01	-5.414459E+01	0.0	0.0000	1.007292E+01	-5.414459E+01	3.210875E+01
0 35		60.0000	-5.000000E-01	-1.008167E+01	-6.019096E+01	0.0	0.0000	-1.008167E+01	-6.019096E+01	2.505465E+01
			5.000000E-01	1.007292E+01	-5.414459E+01	0.0	0.0000	1.007292E+01	-5.414459E+01	3.210875E+01
0 35		90.0000	-5.000000E-01	-1.008167E+01	-6.019096E+01	0.0	0.0000	-1.008167E+01	-6.019096E+01	2.505465E+01
			5.000000E-01	1.007292E+01	-5.414459E+01	0.0	0.0000	1.007292E+01	-5.414459E+01	3.210875E+01

S T R E S S E S I N A X I S - S Y M M E T R I C C O N I C A L S H E L L E L E M E N T S (CCONEAX)

ELEMENT ID.	HARMONIC	POINT ANGLE	FIBRE DISTANCE	STRESSES IN ELEMENT COORD SYSTEM NORMAL-V	NORMAL-U	SHEAR-UV	PRINCIPAL STRESSES (ZERO SHEAR) ANGLE	MAJOR	MINOR	MAXIMUM SHEAR
0 35		135.0000	-5.000000E-01	-1.008167E+01	-6.019096E+01	0.0	0.0000	-1.008167E+01	-6.019096E+01	2.505465E+01
			5.000000E-01	1.007292E+01	-5.414459E+01	0.0	0.0000	1.007292E+01	-5.414459E+01	3.210875E+01
0 35		180.0000	-5.000000E-01	-1.008167E+01	-6.019096E+01	0.0	0.0000	-1.008167E+01	-6.019096E+01	2.505465E+01
			5.000000E-01	1.007292E+01	-5.414459E+01	0.0	0.0000	1.007292E+01	-5.414459E+01	3.210875E+01

1 * * * END OF JOB * * *

EDITOR COMMAND SCRIPT

A file of editor commands which can be used to reformat these analysis results is given below. It is assumed that the analysis results are stored on the file "OUTPUT" and that the editor commands are stored on the file "SCRIPT". The editor would be invoked by the system command:

EDIT, OUTPUT,SCRIPT.

This command makes a working copy of the file "OUTPUT" and positions the current line pointer to the first line. Commands preceded by a number in parentheses are discussed at the end of this section. The results of this editor run are presented in the next section along with a description of the loader commands which have been inserted.

```
(1)  /ENDDATA/,$ D
(2)  X /RINGAX/ D
(3)  G /RINGAX/ C//&@N/
(4)  G /RINGAX/ D
(5)  0 I
     SET,ELEMENTS,INITIAL
     VARIABLE, NAME, MATERIAL, NODE, NODE
     CONSTANT, TYPE, $RING CNICL$
     SET, NODES, INITIAL
     VARIABLE, NAME, X-CORD, Z-CORD, $BAL WT$
     CONSTANT, CORD-SYSTM, 1, Y-CORD, 0, DISP-COORD, 1
     BEGIN DATA
(6)  .
(7)  $ I
     ENDDATA
(8)  .
(9)  \SET\, . K .
(10) \SET\+1 C /NAME/SKIP/
(11) .+1 I
     CONSTANT, NAME, 1000
     INDEX, NAME, 1
     .
(12) W SINPUT
(13) E
     /ENDDATA/,$ D
(14) X /CCONEAX/ D
     G /CCONEAX/ C//&@N/
     G /CCONEAX/ D
     0 I
     SET, ELEMENTS, INITIAL
     VARIABLE, NAME, MATERIAL, NODE, NODE
     CONSTANT, TYPE, $RING CNICL$
     BEGIN DATA
     .
     $ I
```

```
     ENDDATA
     .
     W SINPUT
(15) E
(16) 1,/D I S P L a C E M E N T/ + 4 D
(17) /2  *1  *0.0  *0.0  *0.0  /, $ D
(18) G/  */ C// / G
(19) O I
     SET, NODES
     VARIABLE, NAME SKIP, $X DISP$, $Y DISP$, $Z DISP$
     VARIABLE, X-ROTATION,Y-ROTATION,Z-ROTATION
     BEGIN DATA
     .
     $ I
     ENDDATA
     .
(20) \SET\. . K .
     \SET\+1 C/NAME/SKIP/
     .+1 I
     CONSTANT, NAME, 1000
     INDEX, NAME,1
     .
     W SINPUT
     E
(21) 1,/L O A D   V E C T O R/ + 3 D
(22) /!1/,$ D
(23) G/  */C// /G
     O I
     SET, NODES
     VARIABLE, NAME, SKIP, X-FORCE, Y-FORCE, Z-FORCE
     VARIABLE, X-MOMENT, Y-MOMENT, Z-MOMENT
     BEGIN DATA
     .
     $ I
     ENDDATA
     .
     W SINPUT
     E
     1,/S T R E S S E S/ + 3 D
     .+2, / 35 /-1 D
     .+2,$ D
     G/!?/C// /
     G/  */C// /G
     O I
     SET, ELEMENTS
     VARIABLE, NAME, SKIP, SKIP, $X N STRESS$, $Y N STRESS$, $Z N STRESS$
     VARIABLE, SKIP, $X S STRESS$, $Y S STRESS$, $Z S STRESS$
     BEGIN DATA
     .
     $ I
     ENDDATA
     .
     W SINPUT
     Q
```

The numbered command lines have the following effects:

(1) Delete the first line that contains "ENDDATA" and all subsequent lines in the working file.

(2) Delete all lines except those that contain "RINGAX".

(3) Split each line after the character string "RINGAX".

(4) Delete all lines containing "RINGAX".

(5) Insert the text that follows at the beginning of the file (a "1 B" command could also have been used).

(6) End of text to be inserted.

(7) Insert the text that follows at the end of the file.

(8) End of text to be inserted.

(9) Copy the text from the first previous line containing "SET" through the current line and insert it after the current line (which is also the last line).

(10) The current line is the last line of the file. On the line following the first previous line containing "SET", change "NAME" to "SKIP".

(11) Insert two lines of text after the line following the last line changed.

(12) Write the current contents of the working file to the file "SINPUT", which is being used to accumulate the loader input.

(13) Clear the working file and read a fresh copy of the file "OUTPUT".

(14) As in the commands (1) through (4), retain only the numeric data on the "CCONEAX" lines. Add this data to the loader input file with the necessary header information.

(15) Restore an unedited copy of "OUTPUT".

(16) Delete all lines from the beginning of the file to the start of displacement information.

(17) Starting with the first displacement line that is all zero, delete the remainder of the file. This command begins: Locate the character string which begins with a "2" followed by one or blanks, followed by a "1" followed by one or more blanks, etc.

(18) Replace any string of one or more blanks by just one blank.

(19) Insert header information at the beginning of the file.

(20) Make a copy of the displacement data as was done for the node definitions in commands (9) through (12).

(21) Delete lines preceding the load data from a fresh copy of the file "OUTPUT".
(22) Delete the lines beginning with the new page following the load data through the end of the file. (Most printer output files have special control codes stored in the first character position of each line. Here the character "1" indicates the beginning of a new page. The string "PAGE" could also have been used to identify the same line.)
(23) Delete the first character of each line. In the stress output, a printer control code of "0" has been used to cause double spacing of certain lines. The "go to next line" control code, which is a blank character, appears on most lines and could have been ignored. Here all control codes have been deleted.

LOADER COMMANDS AND LOADER DATA

A file of loader commands and loader data, which can be used to create a STAGING data base for the sample analysis run, is given below. It is assumed that these commands will be stored on the file "SINPUT" and that the loader will be invoked by the system commands:

SLOAD. or

SLOAD,SINPUT.

which are cquivalent.

Both INITIAL and APPEND mode loading are included in this example. A somewhat contrived example of the loader's indexing capability has been included for illustration purposes. The given analysis run would not normally require this feature.

This loader run will create a STAGING data base of the file "TAPE0" which must be retained as a system permanent file for later display by STAGING.

Commands and data lines that are preceded by numbers in parentheses are discussed at the end of this section.

```
(1)  SET,ELEMENTS,INITIAL
     VARIABLE, NAME, MATERIAL, NODE, NODE
     CONSTANT, TYPE, $RING CNICL$
(2)  SET, NODES, INITIAL
(3)  VARIABLE, NAME, X-CORD, Z-CORD, $BAL WT$
(4)  CONSTANT, CORD-SYSTM, 1, Y-CORD, 0, DISP-COORD, 1
```

```
(5)  BEGIN DATA
        2              57.5     .0                  46
        3              58.75    .0                  46
        5              60.      .0                  4
        10             60.      4.                  4
        15             60.      8.                  4
        20             60.      12.                 4
        25             60.      16.                 4
        30             60.      20.                 4
        35             60.      24.                 4
        40             60.      28.                 4
        42             57.5     32.                 46
        43             58.75    32.                 46
        45             60.      32.                 4
        50             60.      36.                 4
        55             60.      40.                 4
        60             60.      44.                 4
        65             60.      48.                 3452
     ENDDATA
(6)  SET, NODES, INITIAL
(7)  VARIABLE, SKIP, X-CORD, Z-CORD, $BAL WT$
     CONSTANT, CORD-SYSTM, 1, Y-CORD, 0, DISP-COORD, 1
(8)  CONSTANT, NAME, 1000
(9)  INDEX, NAME, 1
(10) BEGIN DATA
        2              57.5     .0                  46
        3              58.75    .0                  46
        5              60.      .0                  4
        10             60.      4.                  4
        15             60.      8.                  4
        20             60.      12.                 4
        25             60.      16.                 4
        30             60.      20.                 4
        35             60.      24.                 4
        40             60.      28.                 4
        42             57.5     32.                 46
        43             58.75    32.                 46
        45             60.      32.                 4
        50             60.      36.                 4
        55             60.      40.                 4
        60             60.      44.                 4
        65             60.      48.                 3452
     ENDDATA
(11) SET, ELEMENTS, INITIAL
(12) VARIABLE, NAME, MATERIAL, NODE, NODE
(13) CONSTANT, TYPE, $RING CNICL$
```

```
(14) BEGIN DATA
      2       4       2       3
      3       4       3       5
      5       3       5       10
      10      3       10      15
      15      3       15      20
      20      3       20      25
      25      3       25      30
      30      3       30      35
      35      3       35      40
      40      3       40      45
      42      4       42      43
      43      4       43      45
      45      3       45      50
      50      3       50      55
      55      3       55      60
      60      3       60      65
     ENDDATA
(15) SET, NODES
(16) VARIABLE, NAME, SKIP, $X DISP$, $Y DISP$, $Z DISP$
     VARIABLE, X-ROTATION,Y-ROTATION,Z-ROTATION
(17) BEGIN DATA
      2 0 -9.118936E-05 0.0 -3.630717E-05 0.0 -3.537418E-06 0.0
      3 0 -9.062216E-05 0.0 -3.189937E-05 0.0 -3.515412E-06 0.0
      5 0 -9.010751E-05 0.0 -2.751779E-05 0.0 -3.495443E-06 0.0
      10 0 -1.052826E-04 0.0 -2.556372E-05 0.0 -3.469719E-06 0.0
      15 0 -1.161534E-04 0.0 -2.333901E-05 0.0 -1.917110E-06 0.0
      20 0 -1.210210E-04 0.0 -2.095868E-05 0.0 -6.292796E-07 0.0
      25 0 -1.218593E-04 0.0 -1.852483E-05 0.0 1.275398E-07 0.0
      30 0 -1.202955E-04 0.0 -1.609989E-05 0.0 6.374756E-07 0.0
      35 0 -1.167751E-04 0.0 -1.372601E-05 0.0 1.112742E-06 0.0
      40 0 -1.117314E-04 0.0 -1.143988E-05 0.0 1.272038E-06 0.0
      42 0 -1.095846E-04 0.0 -9.245275E-06 0.0 1.175702E-09 0.0
      43 0 -1.089030E-04 0.0 -9.246740E-06 0.0 1.168388E-09 0.0
      45 0 -1.082845E-04 0.0 -9.248197E-06 0.0 1.161751E-09 0.0
      50 0 -1.117000E-04 0.0 -7.056703E-06 0.0 -1.251518E-06 0.0
      55 0 -1.165694E-04 0.0 -4.772603E-06 0.0 -1.040575E-06 0.0
      60 0 -1.196707E-04 0.0 -2.406617E-06 0.0 -5.029012E-07 0.0
      65 0 -1.206558E-04 0.0 0.0 0.0 0.0 0.0
     ENDDATA
(18) SET, NODES
     VARIABLE, SKIP, SKIP, $X DISP$, $Y DISP$, $Z DISP$
     VARIABLE, X-ROTATION,Y-ROTATION,Z-ROTATION
     CONSTANT, NAME, 1000
     INDEX, NAME, 1
     BEGIN DATA
      2 0 -9.118936E-05 0.0 -3.630717E-05 0.0 -3.537418E-06 0.0
      3 0 -9.062216E-05 0.0 -3.189937E-05 0.0 -3.515412E-06 0.0
      5 0 -9.010751E-05 0.0 -2.751779E-05 0.0 -3.495443E-06 0.0
      10 0 -1.052826E-04 0.0 -2.556372E-05 0.0 -3.469719E-06 0.0
      15 0 -1.161534E-04 0.0 -2.333901E-05 0.0 -1.917110E-06 0.0
```

```
      20 0 -1.210210E-04 0.0 -2.095868E-05 0.0 -6.292796E-07 0.0
      25 0 -1.218593E-04 0.0 -1.852483E-05 0.0 1.275398E-07 0.0
      30 0 -1.202955E-04 0.0 -1.609989E-05 0.0 6.374756E-07 0.0
      35 0 -1.167751E-04 0.0 -1.372601E-05 0.0 1.112742E-06 0.0
      40 0 -1.117314E-04 0.0 -1.143958E-05 0.0 1.272038E-06 0.0
      42 0 -1.095846E-04 0.0 -9.245275E-06 0.0 1.175702E-09 0.0
      43 0 -1.089030E-04 0.0 -9.246740E-06 0.0 1.168388E-09 0.0
      45 0 -1.082845E-04 0.0 -9.248197E-06 0.0 1.161751E-09 0.0
      50 0 -1.117000E-04 0.0 -7.056703E-06 0.0 -1.251518E-06 0.0
      55 0 -1.165694E-04 0.0 -4.772603E-06 0.0 -1.040575E-06 0.0
      60 0 -1.196707E-04 0.0 -2.406617E-06 0.0 -5.029012E-07 0.0
      65 0 -1.206558E-04 0.0 0.0 0.0 0.0 0.0
     ENDDATA
(19) SET, NODES
     VARIABLE, NAME, SKIP, X-FORCE, Y-FORCE, Z-FORCE
     VARIABLE, X-MOMENT, Y-MOMENT, Z-MOMENT
     BEGIN DATA
      5 0 -7.539822E+02 0.0 0.0 0.0 0.0 0.0
      10 0 -1.507964E+03 0.0 0.0 0.0 0.0 0.0
      15 0 -1.507964E+03 0.0 0.0 0.0 0.0 0.0
      20 0 -1.507964E+03 0.0 0.0 0.0 0.0 0.0
      25 0 -1.507964E+03 0.0 0.0 0.0 0.0 0.0
      30 0 -1.507964E+03 0.0 0.0 0.0 0.0 0.0
      35 0 -1.507964E+03 0.0 0.0 0.0 0.0 0.0
      40 0 -1.507964E+03 0.0 0.0 0.0 0.0 0.0
      45 0 -1.507964E+03 0.0 0.0 0.0 0.0 0.0
      50 0 -1.507964E+03 0.0 0.0 0.0 0.0 0.0
      55 0 -1.507964E+03 0.0 0.0 0.0 0.0 0.0
      60 0 -1.507964E+03 0.0 0.0 0.0 0.0 0.0
      65 0 -7.539822E+02 0.0 0.0 0.0 0.0 0.0
     ENDDATA
(20) SET, ELEMENTS
     VARIABLE, NAME, SKIP, SKIP, $X N STRESS$, $Y N STRESS$, $Z N STRESS$
     VARIABLE, SKIP, $X S STRESS$, $Y S STRESS$, $Z S STRESS$
     BEGIN DATA
      15 0 -5.000000E-01 -1.515142E+01 -6.416098E+01 0.0
      5.000000E-01 1.508066E+01 -5.509136E+01 0.0
      35 0 -5.000000E-01 -1.008167E+01 -6.019096E+01 0.0
      5.000000E-01 1.007292E+01 -5.414459E+01 0.0
     ENDDATA
```

The numbered command and data lines have the following effects:

(1) For initial mode loading, all node and element attributes must be defined prior to any actual data loading. This is the beginning of a dummy set that defines the element attributes. The actual data set begins at line (11).

(2) Beginning of the first set of node point data.

(3) Five items are to be read from each data record: The node ID or NAME, the x-coordinate, the y-coordinate, the z-coordinate, and a number indicating the degrees of freedom which are to be constrained at the node. STAGING does not have an attribute name which directly relates to the last quantity, so an unused attribute name, "BAL WT", was chosen to identify the constraints.

(4) All nodal coordinates and displacements are given in the same rectangular coordinate system, "1". (These specifications are mandatory since STAGING has no default coordinate system.) The y-coordinate value is to be set to zero for all nodes.

(5) Node point data given in the format produced by NASTRAN (card names removed).

(6) Beginning of second set of node point data.

(7) This data set is the same as the first set, except that the user will assign node ID's for each data record. Thus the first data field of each record, the original ID, is to be skipped.

(8) Begin the new node ID's with the number 1000.

(9) Increment the new node ID's by one (for nodes 1001 through 1016).

(10) Node point data--identical with line (5).

(11) Beginning of a set of element definitions.

(12) Four items are to be read from each data record: The element ID, the material-type code, and the ID's of the two nodes which the element connects.

(13) All elements are the "RING CNICL" type.

(14) Element data given in the format produced by NASTRAN (card names removed).

(15) Beginning of a set of nodal displacement data. This data must be loaded using the append node since the set of data attributes has changed.

(16) Each data record will contain eight data fields. The second field, which contains the harmonic number (a zero), is to be skipped.
(17) Nodal displacement data given in the format produced by NASTRAN, except that the extra blank characters between data fields have been removed to compress the width of this listing.
(18) Nodal displacement data set for the nodes 1000 through 1016.
(19) Data set of loads to be applied at node points.
(20) Stress data set for elements 15 and 35. The harmonic number and fiber distances are to be skipped. The element stress attribute names are given new interpretations here in order to accommodate the inner and outer stress values produced by NASTRAN's conical shell element. Note that each record is two lines long.

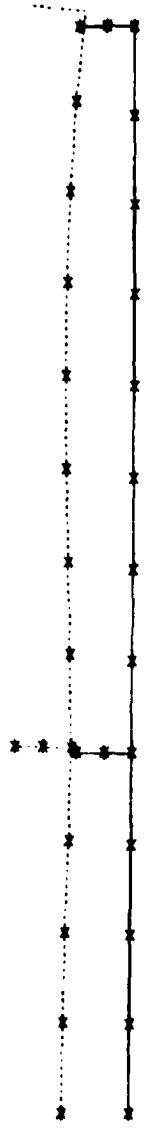
USING STAGING TO DISPLAY MODEL AND ANALYSIS RESULTS

A few STAGING plots have been included in this section to illustrate the type of graphic capabilities one can expect once a data base has been created using the EDITOR/LOADER. Figure 2 is a STAGING summary of the data base contents which lists the various types of data attributes stored in this data base. (The element attribute "BAL WT", included in the loader example, was omitted when this example was prepared.) Figure 3 is a composite plot of the undeformed and the deformed structure. Figure 4 is an x-y plot of the x-displacement of the nodes vs the z-coordinate value of the node.

```
STRUCTURE  COUNT--  1

 NAMED--  STRUCTURE

SSTRUCTURE COUNT--  1

 NAMED--  SSTRUCTURE

ELEMENT    COUNT--  16

 THEY RANGE    FROM 2              TO 60

THE ATTRIBUTES ARE--MATERIAL
                    TYPE
                    X N STRE
                    Y N STRE
                    Z N STRE
                    X S STRE
                    Y S STRE
                    Z S STRE

NODE       COUNT--  17

 THEY RANGE    FROM 2              TO 65

THE ATTRIBUTES ARE--X-CORD
                    Z-CORD
                    CORD-SYS
                    Y-CORD
                    X DISP
                    Y DISP
                    Z DISP
                    X-ROTATI
                    Y-ROTATI
                    Z-ROTATI
                    X-FORCE
                    Y-FORCE
                    Z-FORCE
                    X-MOMENT
                    Y-MOMENT
                    Z-MOMENT
```

```
A. GLOBAL
B. REDRAW
C. ERASE

 CHOOSE GLOBAL COMMAND

D. RESTORE PICTURE
E. COMPACT
F. ENLARGE
G. WINDOW
H. RECENTER

J. SPLIT SCREEN
K. FILL SPACE
L. CHANGE COORD SYSTEM

R. STOP
S. CATALOG AS CURRENT
T. CATALOG AS NEW
U. DATA BASE STATS
V. DISPLAY STATS
W. HELP
X. INTERCOM
Y. RETURN

 CURRENT ATTRIBUTES ARE
    LISTED BY GLOBAL
    COMMAND--DB STATS

D. ADD ATTRIBUTE TO DB
E. RENAME ATTRIBUTE
F. RETURN

X-CORD     FORCE-CORD
Y-CORD     X-ROTATION
Z-CORD     Y-ROTATION
CORD-SYSTMZ-ROTATION
X-FORCE    DISP-COORD
Y-FORCE    TEMPERATUR
Z-FORCE    CONC-MASS
X-MOMENT   PRESSURE
Y-MOMENT   DESCRIPTOR
Z-MOMENT   X DISP

 PICK NODE ATTRIBUTE TO
   BE ADDED, THEN NEXT

D. NEXT
E. RETURN
```

Figure 2 - STAGING Data Base Statistics for the Sample Problem



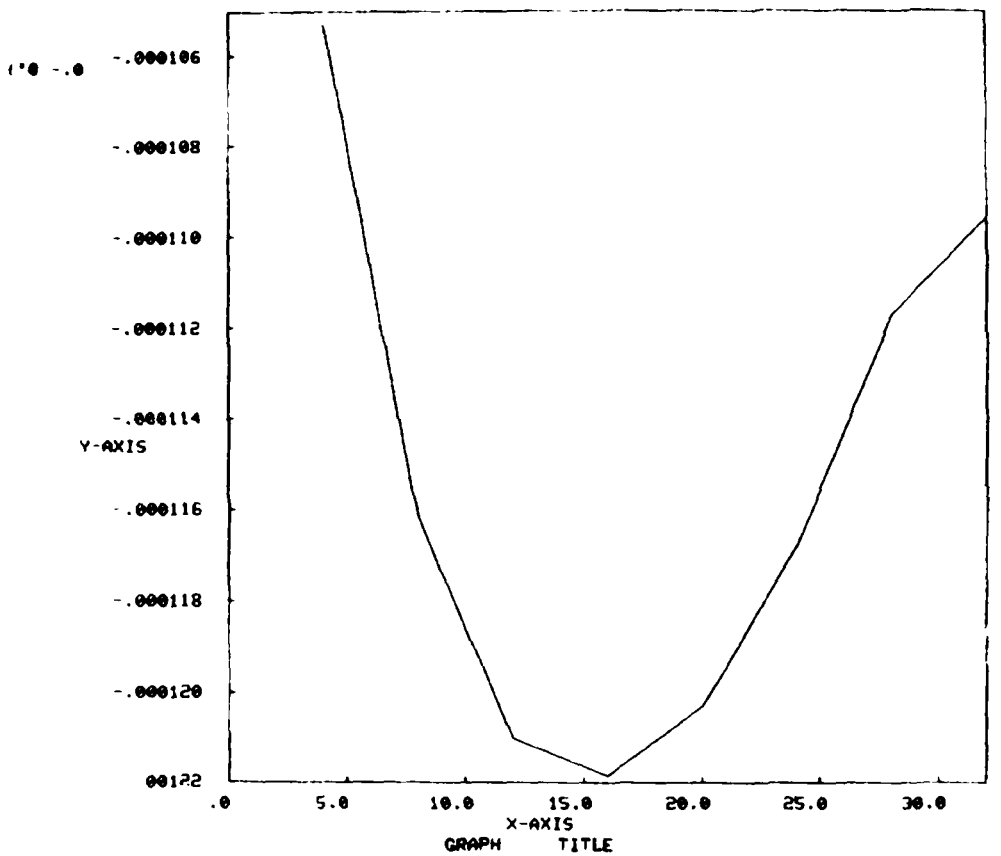
A. GLOBAL
B. ERASE--REDRAW
C. ERASE--NO REDRAW

SET APPROPRIATE
VALUE(S) TO DRAW
DEFORMED PICTURE.

D. SET UNDEFORMED MODE
E. SET SCALE FACTOR

H. BY TIME STEP
I. DRAW DEFORMED PICTUR
J. RETURN

Figure 3 - STAGING Composite Plot of Undeformed and Deformed Structure



A. GLOBAL
B. ERASE--REDRAW
C. ERASE--NO REDRAW

NUMBER	
X-CORD	Y-ROTATION
Z-CORD	Z-ROTATION
CORD-SYSTM	X-FORCE
Y-CORD	Y-FORCE
DISP-COORD	Z-FORCE
X DISP	X-MOMENT
Y DISP	Y-MOMENT
Z DISP	Z-MOMENT
X-ROTATION	

PICK ATTRIBUTE FOR X-AXIS, THEN RETURN

D. RETURN

GENERATE X-Y PLOTS OF NODE ATTRIBUTES

D. ACTIVATE DATA
E. X-AXIS
F. Y-AXIS
G. CHANGE GRAPH
H. CHANGE LINE
I. ERASE--LEAVE ACTIVE
J. ERASE--DEACTIVATE
K. RETURN

NUMBER	
X-CORD	Y-ROTATION
Z-CORD	Z-ROTATION
CORD-SYSTM	X-FORCE
Y-CORD	Y-FORCE
DISP-COORD	Z-FORCE
X DISP	X-MOMENT
Y DISP	Y-MOMENT
Z DISP	Z-MOMENT
X-ROTATION	

PICK ATTRIBUTE FOR Y-AXIS, THEN RETURN

D. RETURN

Figure 4 - STAGING X-Y Plot of X-Displacement of Nodes versus Z-Coorinate Values

LOADING AND MAINTAINING THE PROGRAMS

The STAGING editor and loader programs are written in a FORTRAN language variant called RATFOR which is described in Software Tools.[2] The RATFOR preprocessor translates block-structured FORTRAN code into standard FORTRAN which can be compiled using standard system compilers. Editor and loader program modifications have been partitioned into two phases: subprogram library maintenance and relocatable loading. Subprogram maintenance includes: (1) Source program library changes using the UPDATE utility, (2) RATFOR precompilation and FORTRAN compilation using the FTN optimizing compiler, and (3) Object library maintenance using the EDITLIB utility. Relocatable loading is accomplished using the standard system loader along with appropriate object libraries. Both the editor and the loader require the library DATAMAN which contains the lowest level data base subroutines used by STAGING. All necessary programs and libraries are included on the program distribution tape, which is described in Appendix E.

UPDATING THE TEXT EDITOR

The following control language sequence can be used to modify the STAGING text editor libraries. The source program changes should be prepared in CDC/UPDATE format and stored on a system permanent file named EDITIN. The file EDITPL is the source program library in UPDATE format. The file EDITLB is the object program library in CDC/EDITLIB format.

```
.PROC,EDITUP.
RETURN,EDITIN,OLDPL,NEWPL,RATFOR.
RETURN,EDITPL,COMPILE,FTO,LGO,EDITLB.
FETCH,EDITIN,CAMK.
REQUEST,NEWPL,*PF.
DISCONT,OUTPUT.
ATTACH,OLDPL,EDITPL,ID=CAMK.
UPDATE,N,P,I=EDITIN.
ATTACH,RATFOR.
RATFOR,COMPILE,FTO.
FTN,I=FTO,OPT=2,EL=A.
FETCH,EDITLB,CAMK.
REWIND,DATA.
EDITLIB,I=DATA.
CATALOG,NEWPL,EDITPL,ID=CAMK.
```

```
EXTEND,EDITLB.
PURGE,OLDPL.
PURGE,EDITIN.
EXIT,U.
RETURN,DATA,EDITIN,OLDPL,NEWPL,RATFOR.
RETURN,EDITPL,COMPILE,FTO,LGO,EDITLB.
REVERT.
.DATA,DATA.
LIBRARY(EDITLB,OLD)
REPLACE(*,LGO)
FINISH.
ENDRUN.
```

CREATING AN EXECUTABLE VERSION OF THE EDITOR

The following control language sequence can be used to create an executable version of the editor on the file ED. The main program, EDIT, is explicitly loaded from the EDITLB library to initiate the automatic loading of all other subroutines. The subroutine REMARK is also explicitly loaded to avoid conflicts with a FORTRAN library routine that has the same name.

```
.PROC,EDITLD.
RETURN,EDITLB,ED,DATAMAN.
ATTACH,EDITLB,ID=CAMK.
ATTACH,DATAMAN,CAMVNEWLIB,ID=CAMV.
MAP,PART.
PURGE,ED,ID=CAMK,PW=XR.
LDSET,LIB=EDITLB/DATAMAN.
LIBLOAD,EDITLB,EDIT.
LIBLOAD,EDITLB,REMARK.
NOGO,ED.
CATALOG,ED,ID=CAMK,PW=XR,XR=XR.
RETURN,EDITLB,ED,DATAMAN.
REVERT.
```

UPDATING THE DATA BASE LOADER

The following control language sequence can be used to modify the STAGING data base loader libraries. The source program changes should be prepared in CDC/UPDATE format and stored on a system permanent file named SLOADIN. The file SLOADPL is the source program library in UPDATE format. The file SLOADLB is the object program library in CDC/EDITLIB format.

```
.PROC,SLOADUP.
RETURN,SLOADIN,NEWPL,SLOADPL,COMPILE.
RETURN,RATFOR,FTO,LGO,SLOADLB.
FETCH,SLOADIN,CAMK.
FETCH,SLOADPL,CAMK.
REQUEST,NEWPL,*PF.
UPDATE,P=SLOADPL,N,I=SLOADIN.
ATTACH,RATFOR.
RATFOR,COMPILE,FTO.
FTN,I=FTO,OPT=2,EL=A,R=3.
ATTACH,SLOADLB,ID=CAMK.
REWIND,DATA.
EDITLIB,I=DATA.
CATALOG,NEWPL,SLOADPL,ID=CAMK.
EXTEND,SLOADLB.
DISCARD,SLOADIN.
DISCARD,SLOADPL.
EXIT(U)
RETURN,SLOADIN,NEWPL,SLOADPL,COMPILE.
RETURN,RATFOR,FTO,LGO,SLOADLB,DATA.
REVERT.
.DATA,DATA.
LIBRARY(SLOADLB,OLD)
REPLACE(*,LGO)
FINISH.
ENDRUN.
```

CREATING AN EXECUTABLE VERSION OF THE LOADER

The following control language sequence can be used to create an executable version of the data base loader on the file SLO. The main program, SLOADR, is explicitly loaded from the SLOADLB library to initiate the automatic loading of all other subroutines. The subroutine REMARK is also explicitly loaded to avoid conflicts with a FORTRAN library routine that has the same name. The routines DMINIT and DMSTGT are explicitly loaded from the library DATAMAN to properly position COMMON areas. The libraries SUPLIB and GENERLB contain successively higher level data base manipulation subroutines. The loader and editor use common error message routines which are a part of the EDITLB library.

```
.PROC,SLOADLD.
RETURN,DATAMAN,EDITLB,GENERLB
RETURN,SUPLIB,SLOADLB,SLO.
ATTACH,EDITLB,ID=CAMK,MR=1.
ATTACH,GENERLB,ID=CAMK,MR=1.
```

```
ATTACH,SUPLIB,FELIB,ID=CARK,MR=1.
ATTACH,SLOADLB,ID=CAMK,MR=1.
ATTACH,DATAMAN,CAMVNEWLIB,ID=CAMV.
PURGE,SLO,SLOAD,ID=CAMK,PW=XR.
MAP,PART.
LDSET,LIB=SLOADLB/EDITLB.
LDSET,LIB=GENERLB.
LDSET,LIB=DATAMAN.
LDSET,LIB=SUPLIB.
LIBLOAD,SLOADLB,SLOADR.
LIBLOAD,EDITLB,REMARK.
LIBLOAD,DATAMAN,DMINIT.
LIBLOAD,DATAMAN,DMSTGT.
NOGO,SLO.
CATALOG,SLO,SLOAD,ID=CAMK,PW=XR,XR=XR.
RETURN,DATAMAN,EDITLB,GENERLB.
RETURN,SUPLIB,SLOADLB,SLO.
REVERT.
```


ACKNOWLEDGMENTS

The author wisnes to thank Peter N. Roth for introducing us to the wealth of programs and techniques described in Software Tools.[2] The preliminary work done on the text editor by Mr. Roth and Mr. Melvin E. Haas is greatly appreciated.

APPENDIX A

SUMMARY OF TEXT EDITOR COMMANDS

Summary of commands for the STAGING text editor: EDIT -

```
B  Insert following lines before current line
   <to> B    (lines follow, end with .)

C  Change Text Fragment
   <from>,<to> C <delim><pattern><delim><replacement><delim> <G> <P>
     .   , .   C      <last pattern>                .= last line changed

D  Delete Lines
   <from>,<to> D <N> <P>
     .   , .   D  1                                 .=<from>-1 after delete

E  Edit a File (Clear and Read)
   E <file name>
   E <saved file name>                              .=1 after read

F  Change and/or Display Edit File Name
   F  or  F <file name>
   (<saved file name> is set to <file name> and displayed)    . not changed

G  Apply Command Globally
   <from>,<to> G <delim><pattern><delim> <command>
     1   , $   G     <last pattern>                 . setting depends on <command>
     (applies command to every line that has a pattern match)

I  Input or Insert Lines from the Terminal
   <to> I    (lines follow, end with .)
     .   I                                          .= last line inserted

K  Copy Lines
   <from>,<to> K <to> <P>
     .   , .   K  .                                 .= last line copied

M  Move Lines
   <from>, to> M <to> <P>
     .   , .   M  .                                 .= last line moved

P  Print Lines
   <from>,<to> P
     .   , .   P  1                                 .= last line printed

Q     Quit
      Q

R  Read Lines from a File
   <to> R <file name>
     .   R <saved file name>                        .= last line read

S  Substitute Lines from the Terminal (Delete and Insert)
   <from>,<to> S  (lines follow, end with .)
     .   , .   S                                    .= last line input
```

```
W  Write Lines onto a File
   <from>,<to> W <file name>
     1    , $    W  <saved file name>                . not changed

X  Apply Command Globally (Exclude)
   <from>,<to> X <delim><pattern><delim> <command>
     1    , $    X      <last pattern>                . setting depends on <command>
     (applies command to ever line that does not have a pattern match)

=  Show Line Number
   <to> = <P>
     .    =                                          .=<to> after display

<CR>  Print Next Line  (same as .+1 P )

<P>  Optional Print Line: P means print, omitted means don't
       (The current line after the execution of the command will be printed)

<G>  Optional Global Change: G means global, omitted means not global
       (The change will be applied to all occurances of <pattern> in the line/s)

<delim>  Any desired character (except blank)

<saved file name>  The saved file name is the file name used in the  E  command,
   or the first file name given in an  K  command if  E  is not used, or
   the name given in the last  F  command.

Text Files
   A maximum of six files may be used during one edit session.  All files to
   be used must be included on the execution control card or be one of the
   following default names:  EFILE, F1, F2, F3, F4, SINPUT.  File names
   listed on the execution control card replace corresponding file names
   in the default list.

   All text files will be rewound prior to any read operation.  All text
   files, except the file named SINPUT. will be rewound prior to any write
   operation.

Command Files
   Editor commands can be stored on a file and used instead of the
   connected terminal file to direct the editor.  For a file to be used as
   a command file it must:  be positioned correctly (e.g. rewound), be
   included in the file list, not be the first file in the file list, and
   be named INPUT or SCRIPT.  If file names INPUT and SCRIPT both appear
   in the file list, commands will be read only from the first file listed.
   Editing commands will be read from the command file until an end-of-file
   mark is encountered.  At that point commands will be read from the
   connected terminal file.

Line Numbers    <from> & <to>
   Line numbers may be made up of the following components
      <N>        a decimal number
       0         a non-existant line before the first
       1         the first line
       .         the current line number
       $         the last line
    /<pattern>/  a forward search to $
    \<pattern>\  a backward search to 1
```

The Line Number Components may be combined with + or - or ; to any extent and in any logical order:

```
    .+1   is the next line
    .-1   is the previous line
    $-5   is the 6th line from the end
```

The operator ; resets the value of . and proceeds:

```
    .+1;.+2   is the 3rd line on (same as .+3)
    /ABC/;/ABC/;/ABC/   is the 3rd line on that has ABC
    \ABC\;\EFG\-4   is the 4th line before a line that has an EFG that
        is before the line that has ABC that is before the current line
    1:/ABC/   is the first line in the file that has ABC
    $;\ABC\   is the last line in the file that has ABC
```

Where two line numbers are called for (<from>,<to>), one number can be given to be used for both. If several are given, the last 2 are used. If no line numbers are given, the defaults for the particular command are used.

Search Patterns: <pattern>

Patterns are usually the string of characters within a line that is wanted. A set of characters has been defined to have special meaning in patterns to allow more general searching:

```
     ?          means any character (F??T matches FOOT or FEET or FA2T, etc.)
     !          means the start of the line (if 1st character in pattern)
     $          means the end of the line (if last character in pattern)
   {     }      inclose a class of characters (means "any of these")
                ( {AB} means an A or a B )
                ( {I-N} means any lecter I thru N )
   {^    }      inclose a negated class (means "not any of these")
                ( {^1-9} means any character except a non-zero digit )
     *          means zero or more of the previous
                ( A* means zero or more A's, AA* means 1 or more A's)
                ( {A-Z}{A-Z}* means one or more letters)
                ( ?* means any number of characters, including none)
    @?          means ?
    @!          means !
    @$          means $
    @{          means {
    @@          means @
       etc.
```

A null pattern (<delim><delim>) means to use the last pattern given. A search /ABC/;//;// is the 3rd line on that has ABC. A command /ABC/ C //EFG/ will search for the next line that has ABC and change the ABC to EFG.

Replacement Text: <replacement>

Replacement Text is usually the string of characters that is wanted within the line. A set of characters have been defined to have a special meaning to allow more general replacement:

```
     &          means whatever was matched by the previous search
     @&         means &
     @@         means @
     @N         means that the current line is to be terminated at this
                point and a new line begun
```

Tabs

The editor does not have column tabs, but the indenting of blocks of text can be easily accomplished using global change facilities. First, select a tab character and enter text lines which begin with enough tab characters to indicate the required level of indentation. After the text has been entered, the following command will perform the indenting:

G/<tab character>/C//<blanks to be inserted per tab>/G

Line Truncation

The editor will truncate all long lines to 136 characters with no messages being issued. The editor has no other automatic truncation facility, but the two following global change commands will truncate lines to any length.

```
G/<??...?>/C//&@NTRASH/
G/!TRASH/D
```

where <??...?> is a string of question marks defining the length of the truncated line.

Invoking the Program

After attaching the editor program as file "EDIT" (e.g., FETCH,EDIT,CAMK) the user enters either:

```
EDIT   or
EDIT, <file1>, ..., <file6>, <infile>, <outfile>    ; where
```

<filei> represents the names of optional text files or command files. <infile> is the name of the primary file of editor commands. The default for <infile> is "KEYBRD". <outfile> is the name of the file to which all editor messages are to be written. The default for <outfile> is "SCREEN". The files KEYBRD and SCREEN will be connected as terminal files during an interactive run. The editor program requires about 50000(octal) words of memory for execution.

APPENDIX B

A SYSTEM COMMAND SEQUENCE FOR RUNNING THE TEXT EDITOR

The following command sequence can be used to run the STAGING text editor. This sequence must reside on a file named EDIT and the editor program must reside on a system permanent file named ZZZEDIT. This procedure is invoked with the same system command that is used to run the editor program in the stand-alone mode (e.g., EDIT, <file list>). This procedure sequence performs three functions: (1) If the first file in the file list is a permanent file, that file is copied to the file PFTEMP and passed to the editor program; (2) All files in the file list (actual and default) which are empty (not yet assigned) are assigned to system permanent file units; (3) When the editor program has been terminated, all files which were assigned in (2), but not used by the editor, are deleted.

```
.PROC,EDIT,EFILE=,F1,F2,F3,F4,SINPUT,IN=,OUT=.
RETURN,ZZZEDIT,EDRNDM.
FETCH,ZZZEDIT,CAMK.
IFE(SW1,SW1)
   SWITCH,1.
ENDIF(SW1)
IFE(SW2,SW2)
   SWITCH,2.
ENDIF(SW2)
IFE(SW3,SW3)
   SWITCH,3.
ENDIF(SW3)
IFE(SW4,SW4)
   SWITCH,4.
ENDIF(SW4)
IFE(SW5,SW5)
   SWITCH,5.
ENDIF(SW5)
SET,R16=0.
IFE(.NOT.FILE(F1,AS),ASF1)
   REQUEST,F1,*PF.
   REWIND,F1.
   SWITCH,1.
ENDIF(ASF1)
IFE(.NOT.FILE(F2,AS),ASF2)
   REQUEST,F2,*PF.
   REWIND,F2.
   SWITCH,2.
ENDIF(ASF2)
```

```
IFE(.NOT.FILE(F3,AS),ASF3)
   REQUEST,F3,*PF.
   REWIND,F3.
   SWITCH,3.
ENDIF(ASF3)
IFE(.NOT.FILE(F4,AS),ASF4)
   REQUEST,F4,*PF.
   REWIND,F4.
   SWITCH,4.
ENDIF(ASF4)
IFE(.NOT.FILE(SINPUT,AS),ASSINPUT)
   REQUEST,SINPUT,*PF.
   REWIND,SINPUT.
   SWITCH,5.
ENDIF(ASSINPUT)
SET,R1=FALSE.
IFE($EFILE$.EQ.$$,EEFILE)
   IFE(FILE(#EFILE,AS),ASEFILD)
      IFE(FILE(#EFILE,PF),PFD)
         REWIND,#EFILE.
         RETURN,PFTEMP.
         REQUEST,PFTEMP,*PF.
         REWIND,PFTEMP.
         COPYBF,#EFILE,PFTEMP.
         REWIND,#EFILE,PFTEMP.
         SET,R1=TRUE.
      ENDIF(PFD)
   ELSE(ASEFILD)
      REQUEST,#EFILE,*PF.
      REWIND,#EFILE.
      SET,R1G=1.
   ENDIF(ASEFILD)
ELSE(EEFILE)
   IFE(FILE(EFILE,AS),ASEFILE)
      IFE(FILE(EFILE,PF),PF)
         REWIND,EFILE.
         RETURN,PFTEMP.
         REQUEST,PFTEMP,*PF.
         REWIND,PFTEMP.
         COPYBF,EFILE,PFTEMP.
         REWIND,EFILE,PFTEMP.
         SET,R1=TRUE.
      ENDIF(PF)
   ELSE(ASEFILE)
      REQUEST,EFILE,*PF.
      REWIND,EFILE.
      SET,R1G=2.
   ENDIF(ASEFILE)
ENDIF(EEFILE)
```

```
IFE(R1,ENDRUN)
   ZZZEDIT,PFTEMP,F1,F2,F3,F4,SINPUT,IN,OUT.
ELSE(ENDRUN)
   ZZZEDIT,EFILE,F1,F2,F3,F4,SINPUT,IN,OUT.
ENDIF(ENDRUN)
EXIT(U)
IFE(SW1,S1)
   SWITCH,1.
   IFE(FILE(F1,BOI),QS1)
      RETURN,F1.
   ENDIF(QS1)
ENDIF(S1)
IFE(SW2,S2)
   SWITCH,2.
   IFE(FILE(F2,BOI),QS2)
      RETURN,F2.
   ENDIF(QS2)
ENDIF(S2)
IFE(SW3,S3)
   SWITCH,3.
   IFE(FILE(F3,BOI),QS3)
      RETURN,F3.
   ENDIF(QS3)
ENDIF(S3)
IFE(SW4,S4)
   SWITCH,4.
   IFE(FILE(F4,BOI),QS4)
      RETURN,F4.
   ENDIF(QS4)
ENDIF(S4)
IFE(SW5,S5)
   SWITCH,5.
   IFE(FILE(SINPUT,BOI),QS5)
      RETURN,SINPUT.
   ENDIF(QS5)
ENDIF(S5)
IFE(R1G.EQ.1,S11)
   IFE(FILE(NEFILE,BOI),QS11)
      RETURN,NEFILE.
   ENDIF(QS11)
ENDIF(S11)
IFE(R1G.EQ.2,S12)
   IFE(FILE(EFILE,BOI),QS12)
      RETURN,EFILE.
   ENDIF(QS12)
ENDIF(S12)
RETURN,ZZZEDIT,EDRNDM.
RETURN,KEYBRD,SCREEN.
REVERT.
```

APPENDIX C

SUMMARY OF DATA BASE LOADER COMMANDS

Summary of commands for the STAGING data loader: SLOAD -

BEGIN DATA - Data delimiter for a set of data to be loaded

```
BEGIN DATA
<data record>
   ...
<data record>
ENDDATA
```

A set of data is composed of zero or more data records.

CONSTANT - Define values for attributes which are to be set to the same value for each data record processed

```
CONSTANT, <attr. name>, <value>, ... , <attr. name>, <value>
```

ENDDATA - End-of-run command and end delimiter for a set of data

INDEX - Set increment values for indexing node and element identifiers and subscripted attributes

```
INDEX, <attr. name>, <value>, ... , <attr. name>, <value>
```

If <attr. name> is "NAME" the increment will be applied to the node or element ID. Otherwise, the increment will be applied to the attribute subscript. The default increment is zero.

SET - Specifies loader modes for a new set of data

```
SET,  NODES    , APPEND , PRINT
      ELEMENTS  INITIAL  NOPRINT
```

APPEND - Defines a loader mode which will append data to an existing data base. A copy of the old data base must be available on a file named "TAPE0". APPEND is a default loader mode.

ELEMENTS - Specifies element data loading mode for current data set.

INITIAL - Specifies loader mode which will create a new data base. Data base will be created on a file named "TAPE0". INITIAL mode loading is usually faster than APPEND mode loading. The following restrictions apply to INITIAL mode loading:

- All attributes must be defined for nodes and elements before the first data set is loaded. This necessitates the creation of a dummy set (with no data) to define the attributes for elements (nodes) when node (element) data is to be loaded first. The first BEGIN DATA freezes the definition of attributes for the INITIAL mode. This data set may be null.
- At least one element must be defined even if it is not to be used. The element "TYPE" attribute must also be defined. (these restrictions are imposed by the current versions of STAGING.)

NODES - Specifies node data loading mode for current set. NODES is a default loader mode.

NOPRINT - Inhibits the printing of subsequent command records.

PRINT - Causes printing of each subsequent command record. Data records are not printed. PRINT is a default loader mode.

VARIABLE - Specifies the location of attributes within a data record

VARIABLE, <attr. name>, ..., <attr. name>
SKIP SKIP

The order of the attribute names listed using VARIABLE commands determines the position of the attribute values to be loaded within a data record. SKIP indicates a data field to be skipped.

<attr. name> - The name of an attribute to be referenced

Only the attribute names listed in the Attribute Name Tables in Appendix D will be accepted. Subscripted attributes can be defined using the CONSTANT command and the INDEX command. An attribute name with a zero subscript is treated the same as an attribute name without a subscript. The attribute "NAME" is used to indicate the node or element ID. Attribute values which have embedded blanks must be bracketed by a pair of dollar signs (e.g., $X DISP$).

<data field> - One unit of numeric data

Numeric data within a field must conform to FORTRAN rules for I, F, or E input formats. All data will be interpreted as real numbers and rounded to integer values as required. Default values for null fields are zero. Data fields to be SKIPped must also conform to FORTRAN rules.

<data record> - A unit of data which is to be loaded in the data for one element or node

<data field>, ..., <data field>

A data record may span several lines, each of which may be up to 136 characters long. Data fields within each data record must be separated by blanks or commas. Within a data set each data record must contain the same number of data fields. Each data record begins with a new line.

<value> - A data constant which is either a numeric or character value.

Numeric values must conform to the rules for data fields. Character values which have embedded delimiters (blanks, commas, etc.) must be bracketed by a pair of dollar signs (e.g., $MEM QUAD$).

Data Base Files

For APPEND mode runs, a copy of an existing data base file must be made to a permanent file device prior to running the STAGING loader. The data base file must be named "TAPE0". After the loader has completed execution, the file "TAPE0" should be cataloged as a permanent file. For INITIAL mode runs, the user needs only to catalog the file "TAPE0" after the run has completed.

Invoking the Loader

After attaching the loader program as file "SLOAD" (e.g., FETCH,SLOAD,CANK), the user enters either:

```
SLOAD    or
SLOAD, <infile>    ; where
```

<infile> is the name of the file which contains the loader commands and data. This file will be rewound at the beginning of the job. The default for the file name is "SINPUT". Messages issued by the loader are written to the file "OUTPUT". The file OUTPUT will be connected as a terminal file during an interactive run. The loader will terminate whenever the data is exhausted on <infile> or whenever an ENDDATA command is encountered outside of a BEGIN DATA/ENDDATA data set. The loader program requires about 50000(octal) words of memory for execution.

Command Format

A command line can be up to 136 characters in length. Commands are composed of alphanumeric fields which are delimited by commas or blanks. Commands may not be continued beyond one line, but two or more commands of the same type can be used to specify a large number of command fields.

APPENDIX D

STAGING DATA ATTRIBUTE NAMES

The following tables have been reproduced from the STAGING USER'S GUIDE[3] and include all currently defined data attributes.

TABLE 1 - STAGING NODE ATTRIBUTES

No.	Full Name	Attributes	Abbreviation	Values
1	X Coordinate	X-CORD	X	any[1]
2	Y Coordinate	Y-CORD	Y	any[1]
3	Z Coordinate	Z-CORD	Z	any[1]
4	Coordinate System	CORD-SYSTM	COS	1=Cartesian 2=Polar(2D)Cylindrical(3D) 3=Spherical(3D Only)
5	X Force	X-FORCE	XFO	any
6	Y Force	Y-FORCE	YFO	any
7	Z Force	Z-FORCE	ZFO	any
8	X Moment	X-MOMENT	XMO	any
9	Y Moment	Y-MOMENT	YMO	any
10	Z Moment	Z-MOMENT	ZMO	any
11	Force Coordinate System	FORCE-CORD	FCO	Local Cord Sys. ID
12	Balance Weight	BAL WT	BWT	any
13	X Rotation	X-ROTATION	XRO	any
14	Y Rotation	Y-ROTATION	YRO	any
15	Z Rotation	Z-ROTATION	ZRO	any
16	Displacement Coordinate System	DISP-COORD	DCO	Local Coord Sys. ID
17	Temperature	TEMPERATUR	TEM	any

TABLE 1 (Continued)

No.	Full Name	Attributes	Abbreviation	Values
18	Concentrated Mass	CONC-MASS	CMS	any
19	Pressure	PRESSURE	PRE	any
20	X Displacement	X DISP	XDS	any[2]
21	Y Displacement	Y DISP	YDS	any[2]
22	Z Displacement	Z DISP	ZDS	any[2]
23	X Mode Shape	X MODE	XMD	any[2]
24	Y Mode Shape	Y MODE	YMD	any[2]
25	Z Mode Shape	Z MODE	ZMD	any[2]
26	X Load	X LOAD	XLD	any[2]
27	Y Load	Y LOAD	YLD	any[2]
28	Z Load	Z LOAD	ZLD	any[2]

[1]Interpretation of X, Y, Z varies according to Attribute 4. If CORD-SYSTM is 1.0 (Cartesian), X, Y, Z are coordinates in space. If CORD-SYSTM is 2.0 and only X and Y are provided (Polar), Attribute 1 is R and Attribute 2, θ measured in radians. If X, Y, Z are provided (cylindrical), Attribute 3 is Z as in the Cartesian case. If X, Y, Z are provided and CORD-SYSTM is 3.0 (spherical), Attribute 1 is R, 2, θ (radians) and 3, ϕ (radians).

[2]The displacement factors are delta displacements from the node Point X, Y, A and must be provided in the same coordinate system. The delta values can refer to up to ten mode shapes, load conditions, or time steps. Only one of these conditions (mode shapes, load conditions, or time steps) can be active at any one time in the data base.

TABLE 2 - STAGING ELEMENT ATTRIBUTES

No.	Full Name	Attribute Name	Abbreviation
1	Type	TYPE *	TYP
2	Material Identifier	MATERIAL	MAT
3	Area Cross-Section	AREA-CRSST	CSA
4	Area Moment X**Direction	X AREA MOM	XAM
5	Area Moment Y Direction	Y AREA MOM	YAM
6	Area Moment Z Direction	Z AREA MOM	ZAM
7	Torsional Constant	TORSIONAL	TOR
8	Mass/Length	MASS/LENGT	MPL
9	Membrane Thickness	MEM THICK	MTH
10	Mass/Area	MASS/AREA	MPA
11	Flexual Thickness	FLEX THICK	FTH
12	Material Property A	MAT-PROP-A	PRA
13	Pressure	PRESSURE	PRE
14	Temperature	TEMPERATUR	TEM
15	Critical Load	CRIT LOAD	CLD
16	Design Criterion	DES CRIT	DCR
17	Construction Code	CONSTRCODE	CCD
18	Geometry Class	GEOMCLASS	GCL
19	Geometry Sub-Class	SGEOMCLASS	SGC
20	Angle Between Prop-Axes & Side I-T	BETA	BET
21	Tension Allowable Stress	TEN ALWSTR	TAL
22	Compression Allowable Stress	CMP ALWSTR	SAL
23	Shear Allowable Stress	SHR ALWSTR	SAL
24	Minimum Size	MIN SIZE	MIN
25	Maximum Size	MAX SIZE	MAX
26	Allowable Class	ALOWCLASS	ALC
27	Allowable Sub-Class	SALOWCLASS	SCN
28	Average Stress Concentration Ratio	STRCNSTR	STC
29	Original Thickness	ORIG THICK	OTH
30	Excluded Element	EXCLUD ELM	EXE
31	Non-Optimum Weight Factor	NOPTWTFAC	NPW
32	Normal Stress X (Centroid)	X N STRESS	XNS
33	Normal Stress Y (Centroid)	Y N STRESS	YNS
34	Normal Stress Z (Centroid)	Z N STRESS	ZNS
35	Shear Stress X (Centroid)	X S STRESS	XSS
36	Shear Stress Y (Centroid)	Y S STRESS	YSS
37	Shear Stress Z (Centroid)	Z S STRESS	ZSS
38	Maximum Principal Stress	MAX STRESS	MXS
39	Intermediate Principal Stress	INT STRESS	INS
40	Minimum Principal Stress	MIN STRESS	MNS
41	Equivalent Stress	EQU STRESS	EQS

*See Table 3.

**XYZ--GLOBAL Cartesian Coordinate System.

TABLE 3 - STAGING ELEMENT TYPES

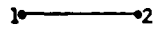
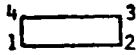
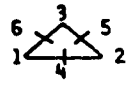
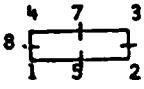
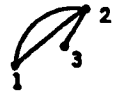
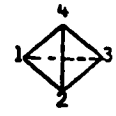
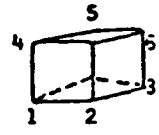
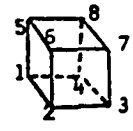
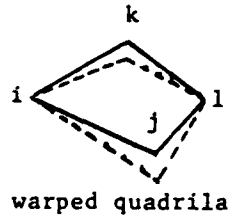
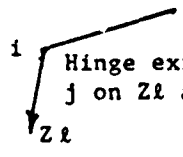
TYPE NO.		FULL NAME	SHORTHAND NAME
1		Rod	ROD
2		Straight Beam	STR BEAM
3		Membrane Triangle	MEM TRIA
4		Membrane Quadrilateral	MEM QUAD
5	Same as 3	Plate Triangle	PLATE TRIA
6	Same as 4	Plate Quadrilateral	PLATE QUAD
7	Same as 3	General Triangle	GEN TRIA
8	Same as 4	General Quadrilateral	GEN QUAD
9	Same as 3	Ring Triangle	RING TRIA
10	Same as 4	Ring Quadrilateral	RING QUAD
11	Same as 4	Shear Panel	SHER PANEL
12	Same as 4	Twist Panel	TWIS PANEL
13		General Triangle(2)	GEN TRIA2
14		General Quadrilateral(2)	GEN QUAD2
15	Same as 13	Ring Triangle(2)	RING TRIA2
16	Same as 14	Ring Quadrilateral(2)	RING QUAD2
17		Curved Beam	CURVE BEAM
18	Same as 17	Ring Shell	RING SHELL
19		Ring Conical	RING CNICL

TABLE 3 (Continued)

TYPE NO.		FULL NAME	SHORTHAND NAME
20		Tetrahedral Solid	TETR SOLID
21		Wedge Solid	WEDG SOLID
22		Hexahedral Solid	HEXA SOLID
23		20 Node Brick	BRICK20
24	1—AS—2	Axial Spring	AXIAL SPRG
25	1—TS—2	Torsional Spring	TORSN SPRG
26	1—M—2	Mass	MASS
27	1—D—2	Damper	DAMPER
28	 warped quadrilateral	Warped Quadrilateral	WARP QUAD
29	 Hinge exists at node j on Zℓ axis	Hinged Beam	HINGE BEAM

APPENDIX E

PROGRAM DISTRIBUTION TAPE DESCRIPTION

The text editor and loader programs are distributed on an 800 BPI, 7-track tape produced in SCOPE format on DTNSRDC'S CDC-NOS/BE operating system. There are 20 files on the tape arranged as follows:

File Number	File Name	Contents	Format
1.	EDIT	Editor Program - Executable	Binary
2.	SLOAD	Loader Program - Executable	Binary
3.	TESTOUT	Example NASTRAN Output file	Text
4.	SINPUT	Loader Input for Small Example	Text
5.	SINPUTX	Loader Input for NASTRAN Example	Text
6.	SCRIPT	Editor Commands for NASTRAN Example	Text
7.	INDEXTEST	Loader Input for Indexing Example	Text
8.	EDITUP	Editor Updating Command Sequence	Text
9.	EDITLD	Editor Loading Command Sequence	Text
10.	SLOADUP	Loader Updating Command Sequence	Text
11.	SLOADLD	Loader loading Command Sequence	Text
12.	RATFOR	RATFOR Precompiler - Executable	Binary
13.	EDITPL	Editor Source Program Library	UPDATE Seq.
14.	SLOADPL	Loader Source Program Library	UPDATE Seq.
15.	RATPL	GENERLB Source Program Library	UPDATE Seq.
16.	EDITLB	Editor Object Program Library	EDITLIB Seq.
17.	SLOADLB	Loader Object Program Library	EDITLIB Seq.
18.	GENERLB	Data Base and Utility Library	EDITLIB Seq.
19.	DATAMAN	Low Level Data Base Library	EDITLIB Seq.
20.	SUPLIB	Intermediate Data Base Library	EDITLIB Seq.

All programs and control sequences are distributed as they are currently used on DTNSRDC'S CDC 6000 computer systems. As such, there may be minor variations from the listings that appear in the report.

REFERENCES

1. THE STAGING SYSTEM: VOLUME II - USER'S GUIDE, Battelle Columbus Laboratories Report to the Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio (28 April 1979).

2. Kernighan, B.W. and P.J. Plauger, Software Tools, Addison-Wesley, Reading, Massachusetts (1976).

3. THE NASTRAN USER'S MANUAL, NASA SP-222(4), Washington, D.C. (December 1977).

INITIAL DISTRIBUTION

Copies	
1	USA PICATINNY ARSENAL
	1 B. Nagel
1	NRL
	1 R. Perlut
1	DNL
2	USNA
	1 Dept Math
	1 Tech Lib
1	NAVPGSCOL/Lib
1	NAVWARCOL
1	ROTC, MIT
1	NSWC/White Oak
	1 R.J. Edwards
1	NUSC/New London
	1 A. Carlson
1	NAVSEA
	1 SEA 321/J. Claffey
1	NAVAIR
	1 G. Hand
1	NAVSHIPYD BREM/Lib
1	NAVSHIPYD CHASN/Lib
1	NAVSHIPYD MARE/Lib
1	NAVSHIPYD NORVA/Lib
1	NAVSHIPYD PEARL/Lib
1	NAVSHIPYD PHILA/Lib
1	NAVSHIPYD PTSMH/Lib
12	DTIC

Copies	
15	AFFDL
	12 B. Groomes
	1 P. Pourier
	1 J. Johnson
	1 J. Folck
1	NASA Goddard SFC
	1 J. Mason
1	BATTELLE COLUMBUS LABS
	505 King Avenue
	Columbus, Ohio 43201
	1 Gene Hulbert

Center Distribution

Copies	Code	Name
1	1182	Z.G. Wachnik
1	1568	L. Motter
1	17	W.W. Murray
1	1720.2	K. Hom
1	1720.6	R.D. Rockwell
1	1730	A.B. Stavovy
2	1730.5	J.C. Adamchak J. Figula
2	1740.5	P. Roth B. Wang
1	1800	G.H. Gleissner
1	1802.1	H.J. Lugt
1	1805	E.H. Cuthill
2	1809.3	D. Harris
1	1824	S. Berkowitz
1	1840	J.W. Schoc
2	1843	H.J. Haussling M.E. Haas

Copies	Code	Name
1	1844	S.K. Dhir
25	1844	J.M. McKee
1	1892.1	J. Strickland
1	1892.2	D. Sommer
1	1966	J. Caspar
10	5211.1	Reports Distribution
1	522.1	Unclassified Lib (C)
1	522.2	Unclassified Lib (A)